

Problem 1

IntVector.cpp

```
#include "IntVector.h"
#include <iostream>

using namespace std;

IntVector::IntVector(const int aArrayOfIntegers[], size_t aNumberOfElements)
{
    fNumberOfElements = aNumberOfElements;
    fElements = new int[fNumberOfElements];

    for (size_t i = 0; i < fNumberOfElements; i++) {
        fElements[i] = aArrayOfIntegers[i];
    }
}

IntVector::~IntVector()
{
    delete fElements;
}

size_t IntVector::size() const {
    return fNumberOfElements;
}

void IntVector::swap(size_t aSourceIndex, size_t aTargetIndex) {
    if (aSourceIndex == aTargetIndex) {
        throw out_of_range("Can't swap the same index.");
    }

    if (aSourceIndex >= fNumberOfElements) {
        throw out_of_range("Illegal Source Vector Index.");
    }

    if (aTargetIndex >= fNumberOfElements) {
        throw out_of_range("Illegal Target Vector Index.");
    }

    int lBuffer = fElements[aSourceIndex];
    fElements[aSourceIndex] = fElements[aTargetIndex];
    fElements[aTargetIndex] = lBuffer;
}

void IntVector::sort(const IntSorter& aSorter) {
    aSorter(*this);
}

const int IntVector::operator[](size_t aIndex) const {
    if (aIndex < 0 || aIndex >= fNumberOfElements) {
        throw out_of_range("Illegal Vector Index.");
    }

    return fElements[aIndex];
}

IntVectorIterator IntVector::begin() const {
    return IntVectorIterator (*this);
}

IntVectorIterator IntVector::end() const {
    return IntVectorIterator(*this, size());
}
```

IntVectorIterator.cpp

```
#include "IntVector.h"
#include "IntVectorIterator.h"

IntVectorIterator::IntVectorIterator(const IntVector& aContainer, size_t aStart) :
fContainer(aContainer), fPosition(aStart) {

}

const int IntVectorIterator::operator*() const {
    return fContainer[fPosition];
}

IntVectorIterator& IntVectorIterator::operator++() {
    fPosition++;
    return *this;
}

IntVectorIterator IntVectorIterator::operator++(int) {
    IntVectorIterator old = *this;
    ++(*this);
    return old;
}

bool IntVectorIterator::operator==(const IntVectorIterator& aRHS) const {
    return
        &fContainer == &aRHS.fContainer &&
        fPosition == aRHS.fPosition;
}

bool IntVectorIterator::operator!=(const IntVectorIterator& aRHS) const {
    return !(*this == aRHS);
}

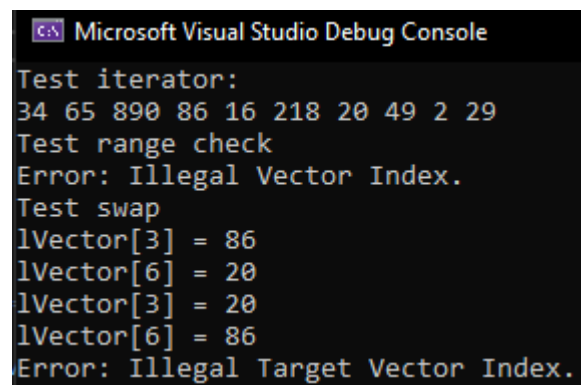
IntVectorIterator IntVectorIterator::begin() const {
    IntVectorIterator iter = *this;

    iter.fPosition = 0;
    return iter;
}

IntVectorIterator IntVectorIterator::end() const {
    IntVectorIterator iter = *this;

    iter.fPosition = iter.fContainer.size();
    return iter;
}
```

Output



```
Microsoft Visual Studio Debug Console
Test iterator:
34 65 890 86 16 218 20 49 2 29
Test range check
Error: Illegal Vector Index.
Test swap
lVector[3] = 86
lVector[6] = 20
lVector[3] = 20
lVector[6] = 86
Error: Illegal Target Vector Index.
```

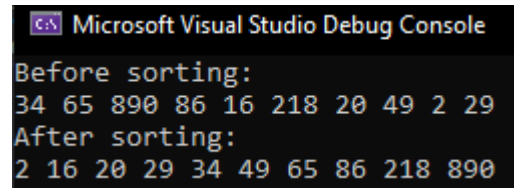
CocktailShakerSort.cpp

```
#include "IntVector.h"
```

```
#include "CocktailShakerSort.h"
```

```
void CocktailShakerSort::operator()(IntVector& aContainer) const {  
  
    int beginIndex = 0;  
    int endIndex = aContainer.size() - 1;  
  
    while (beginIndex < endIndex) {  
        for (int i = beginIndex; i <= endIndex - 1; i++)  
        {  
            if (aContainer[i] > aContainer[i + 1]) {  
                aContainer.swap(i, i + 1);  
            }  
        }  
  
        endIndex--;  
  
        for (int i = endIndex; i >= beginIndex + 1; i--)  
        {  
            if (aContainer[i] < aContainer[i - 1]) {  
                aContainer.swap(i, i - 1);  
            }  
        }  
  
        beginIndex++;  
    }  
}
```

Output



Microsoft Visual Studio Debug Console

```
Before sorting:  
34 65 890 86 16 218 20 49 2 29  
After sorting:  
2 16 20 29 34 49 65 86 218 890
```